部署LNMP环境（linux,nginx,mysql,php）A=apache

操作过程中需要安装的软件列表如下：

nginx

mariadb、mariadb-server、mariadb-devel

php、php-fpm、php-mysql

备注：mariadb（数据库客户端软件）、mariadb-server（数据库服务器软件）、mariadb-devel（其他客户端软件的依赖包）、php（解释器）、php-fpm（进程管理器服务）、php-mysql（PHP的数据库扩展包）。

**步骤一：安装软件**

**1.使用yum安装基础依赖包**

[root@proxy ~]# yum -y install gcc openssl-devel pcre-devel zlib-devel

**2.源码安装Nginx**

[root@proxy ~]# useradd -s /sbin/nologin nginx

[root@proxy ~]# tar -xvf nginx-1.12.2.tar.gz

[root@proxy ~]# cd nginx-1.12.2

[root@proxy nginx-1.12.2]# ./configure \

> --user=nginx --group=nginx \

> --with-http\_ssl\_module

[root@proxy ~]# make && make install

**3.安装MariaDB**

[root@proxy ~]# yum -y install mariadb mariadb-server mariadb-devel

**4.php和php-fpm(该软件包在lnmp\_soft中提供)cd到lnmp\_soft安装**

[root@proxy ~]# yum -y install php php-mysql

[root@proxy ~]# yum -y install php-fpm-5.4.16-42.el7.x86\_64.rpm

**步骤二：启动服务**

1）启动Nginx服务（如果已经启动nginx，则可以忽略这一步）

这里需要注意的是，如果服务器上已经启动了其他监听80端口的服务软件（如httpd），则需要先关闭该服务，否则会出现冲突。

[root@proxy ~]# systemctl stop httpd //如果该服务存在则关闭该服务

[root@proxy ~]# /usr/local/nginx/sbin/nginx //启动Nginx服务

[root@proxy ~]# netstat -utnlp | grep :80

2）启动MySQL服务

[root@proxy ~]# systemctl start mariadb //启动服务器

[root@proxy ~]# systemctl status mariadb //查看服务状态

[root@proxy ~]# systemctl enable mariadb //设置开机启动

3）启动PHP-FPM服务

[root@proxy ~]# systemctl start php-fpm //启动服务

[root@proxy ~]# systemctl status php-fpm //查看服务状态

[root@proxy ~]# systemctl enable php-fpm //设置开机启动

1. 设置防火墙与SELinux

[root@proxy ~]# firewall-cmd --set-default-zone=trusted

[root@proxy ~]# setenforce 0

## **构建LNMP平台**

1.查看php-fpm配置文件（实验中不需要修改该文件）

[root@proxy etc]# vim /etc/php-fpm.d/www.conf

[www]

listen = 127.0.0.1:9000 //PHP端口号

pm.max\_children = 32 //最大进程数量

pm.start\_servers = 15 //最小进程数量

pm.min\_spare\_servers = 5 //最少需要几个空闲着的进程

pm.max\_spare\_servers = 32 //最多允许几个进程处于空闲状态

**2.修改Nginx配置文件并启动服务**

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

location / {

root html;

index index.php index.html index.htm;

#设置默认首页为index.php，当用户在浏览器地址栏中只写域名或IP，不说访问什么页面时，服务器会把默认首页index.php返回给用户

}

location ~ \.php$ { #匹配末尾是.php结尾的文件

root html;

fastcgi\_pass 127.0.0.1:9000; #将请求转发给本机9000端口，PHP解释器

fastcgi\_index index.php;

#fastcgi\_paramSCRIPT\_FILENAME/scripts$fastcgi\_script\_name;

include fastcgi.conf; 修改

}

[root@proxy ~]# /usr/local/nginx/sbin/nginx -s reload

可能出现的错误：

1. 没有test.php文件

File not found

1. nginx+php 没有启动

无法连接 An error

3.test.php 语法

4下载（location动静分离）

网页没有任何提示错误，查看日志排错

# tailf /var/log/php-fpm/www-error.log

3.创建PHP页面，测试LNMP架构能否解析PHP页面

[root@proxy ~]# vim /usr/local/nginx/html/test1.php

创建PHP测试页面,连接并查询MariaDB数据库

[root@proxy ~]# cd /root/lnmp\_soft/php\_scripts/ 其下的mysql.php复制到test2.php

[root@proxy ~]# vim /usr/local/nginx/html/test2.php

客户端使用浏览器访问服务器PHP首页文档，检验是否成功：

[root@client ~]# firefox http://192.168.4.5/test1.php

4）LNMP常见问题

Nginx的默认访问日志文件为/usr/local/nginx/logs/access.log

Nginx的默认错误日志文件为/usr/local/nginx/logs/error.log

PHP默认错误日志文件为/var/log/php-fpm/www-error.log

如果动态网站访问失败，可用参考错误日志，查找错误信息。

## **地址重写**

修改配置文件(访问a.html重定向到b.html) 旧地址匹配正则

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

server {

listen 80;

server\_name localhost;

location / {

root html;

index index.html index.htm;虚拟主机

/usr/local/nginx/conf/nginx.conf

server{

listen 80;

server\_name www.a.com;

auth\_basic “123:”;

auth\_basic\_user\_file “/usr/local/ningx/pass”;

root html;

index index.html index.htm

}

rewrite /a.html /b.html; 添加这行

2）重新加载配置文件

[root@proxy ~]# echo "BB" > /usr/local/nginx/html/b.html

3）客户端测试

[root@client ~]# firefox http://192.168.4.5/a.htm

(rewrite /a.html /b.html redirect; 在后面加“redirect”,页面内容是一样的，地址栏地址就是/b.html)

**修改配置文件(访问192.168.4.5的请求重定向至www.tmooc.cn**

1) 修改Nginx服务配置

server {

listen 80;

server\_name localhost;

**rewrite ^/ <http://www.tmooc.cn/;> 另外添加这行，从根开始**

location / {

root html;

index index.html index.htm;

# rewrite /a.html /b.html redirect;

2）重新加载配置文件

3）真实机测试，真实机才可以连接tmooc）

**修改配置文件(访问192.168.4.5/下面子页面，重定向至www.tmooc.cn/下相同的页面)**

1) 修改Nginx服务配置

server {

listen 80;

server\_name localhost;

rewrite ^/(.\*)$ <http://www.tmooc.cn/$1;> 修改此项

location / {

root html;

index index.html index.htm;

# rewrite /a.html /b.html redirect;

5.修改配置文件(实现curl和火狐访问相同链接返回的页面不同)

1) 创建网页目录以及对应的页面文件：

[root@proxy ~]# echo "I am Normal page" > /usr/local/nginx/html/test.html

[root@proxy ~]# mkdir -p /usr/local/nginx/html/firefox/

[root@proxy ~]# echo "firefox page" > /usr/local/nginx/html/firefox/test.html

2) 修改Nginx服务配置

server {

listen 80;

server\_name localhost;

location / {

root html;

index index.html index.htm;

}

#这里，~符号代表正则匹配，\*符号代表不区分大小写

if ($http\_user\_agent ~\* firefox) { //识别客户端firefox浏览器

rewrite ^(.\*)$ /firefox/$1;

}

3.客户端测试

[root@client ~]# firefox http://192.168.4.5/test.html

[root@client ~]# curl http://192.168.4.5/test.html

5）地址重写格式【总结】

rewrite 旧地址 新地址 [选项];

last 不再读其他rewrite rewrite /a.tst /b.tst last;

rewrite /b.txt /c.tst;

break 不再读其他语句，结束请求 location {

rewrite /a.tst /b.tst break;

location{

rewrite /b.tst /c.tst;

redirect 临时重定向

permament 永久重定向

**总结：**

1. 部署LNMP环境
2. 软件安装

源码安装nginx

gcc pcre-devel openssl-devel zlib-devel

mariadb mariadb-server mariadb-devel

php php-mysql

cd nginx-1.12.2

yum -y install php-fpm-5.4.16-42.el7.x86\_64.rpm

**2.重启服务，若开启了httpd,则先关闭httpd,再启动nginx**

systemctl stop httpd

nginx

netstat -anputl | grep :80

systemctl start mariadb

systemctl status mariadb

systemctl enable mariadb

systemctl start php-fpm

systemctl status mariadb

systemctl enable mariadb

**二．构建LNMP平台**

查看php-fpm配置文件

/etc/php-fpm.d/www.conf

1. **修改nginx配置并启动服务**

locatiion ~ /.php${ #打开location括号内的并修改下面内容

**.....**

#fastcgi\_param...

include fastcgi.conf

}

**步骤三：创建PHP页面，测试LNMP架构能否解析PHP页面**

1.创建php页面放到/usr/local/nginx/html/test1.php

cp /root/lnmp\_soft/php\_scripts/mysql.php /usr/local/nginx/html/test2.php

**2.测试**：firefox <http://192.168.4.5/test.php>

firefox http://192.168.4.5/test.php

**三．地址重写（访问a.html重定向到b.html）**

1.修改配置文件/usr/local/nginx/conf/nginx.conf

server {

.....

location / {

index index.html index.htm；

rewrite /a.html /b.html （redirect）；

}

echo “hello world” > /usr/local/nginx/html/index.html

nginx -s reload

2.测试：firefox http://192.168.4.5/a.html

**四．修改配置文件(访问192.168.4.5的请求重定向至www.tmooc.cn)**

1. **修改配置文件**

server {

**......**

**rewrite ^/ <http://www.tmooc.cn/;>**

location /{

2.测试：firefox http://192.168.4.5

**修改配置文件（访问192.168.4.5/下面子页面，重定向到www.tmooc.cn/下相同页面）**

**1.修改配置文件**

server {

......

**rewrite ^/(.\*)$ <http://www.tmooc.cn/$1;>**

location / {

**......**

#rewrite /a.html /b.html;

}

**2.测试**：firefox 192.168.4.5/free free 为www.tmooc.cn下面的子页面

**五．.实现curl和火狐访问相同链接返回的页面不同**

echo “I am page”> /usr/local/nginx/html/test.html

echo “firefox page”> /usr/local/nginx/html/test.html

1. 修改配置文件

location {

......

if ($http\_user\_agent ~\* firefox) { #识别客户端firefox浏览器 注意空格

rewrite ^(.\*) /firefox/$1;

}

测试： firefox <http://192.168.4.5/test.html>

curl <http://192.168.4.5/test.html>

nginx开机需要重启，可以把重启命令/usr/local/nginx/sbin/nginx放在脚本里面

开机运行的脚本在 /etc/rc.local----/etc/rc.d/rc.local 是同一个

先给/etc/rc.d/rc.local 加个执行权限

chmod +x /etc/rc.d/rc.local

**日志文件：/usr/local/nginx/logs/access.log**

192.168.4.5 - - [30/Oct/2018:14:09:43 +0800] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5.0 (X11; Linux x86\_64; rv:52.0) Gecko/20100101 Firefox/52.0"

哪个ip地址用户什么时间浏览了什么页面 $status状态 $body\_bytes访问的页面多少字节 $http\_referer是从哪个网页跳过来的

$http\_user\_agent 代理变量名，从“Mozilla.......Firefox/52.0”都是，知道对方用的什么系统，什么浏览器访问的